

# Theoretische Grundlagen der Informatik

## Übung

9. Übungstermin · 09. Februar 2017  
Marcel Radermacher

INSTITUT FÜR THEORETISCHE INFORMATIK · LEHRSTUHL ALGORITHMIK

# Übersicht

## Inhalt

- Überblick
- Typ-1 Grammatik
- NP-Vollständigkeit & Approximation
- Reduktionsschemas
- Klausurhinweise



# Chomsky-Hierarchien

Menge aller Sprachen

Typ 0  
Rekursiv aufzählbar

Typ 1  
Kontextsensitiv

Typ 2  
Kontextfrei

Typ 3  
Regulär

# Einordnung

Typ	Beschreibungsmittel
0	
1	
2	
3	

# Einordnung

Typ	Beschreibungsmittel
0	<ul style="list-style-type: none"><li>• TM</li><li>• 'Typ-0 Grammatik'</li></ul>
1	
2	
3	

# Einordnung

Typ	Beschreibungsmittel
0	<ul style="list-style-type: none"><li>● TM</li><li>● 'Typ-0 Grammatik'</li></ul>
1	<ul style="list-style-type: none"><li>● linear beschränkte TM (LBA)</li><li>● kontextsensitive Grammatik</li></ul>
2	
3	

# Einordnung

Typ	Beschreibungsmittel
0	<ul style="list-style-type: none"><li>• TM</li><li>• 'Typ-0 Grammatik'</li></ul>
1	<ul style="list-style-type: none"><li>• linear beschränkte TM (LBA)</li><li>• kontextsensitive Grammatik</li></ul>
2	<ul style="list-style-type: none"><li>• PDA</li><li>• kontextfreie Grammatik</li></ul>
3	

# Einordnung

Typ	Beschreibungsmittel
0	<ul style="list-style-type: none"><li>• TM</li><li>• 'Typ-0 Grammatik'</li></ul>
1	<ul style="list-style-type: none"><li>• linear beschränkte TM (LBA)</li><li>• kontextsensitive Grammatik</li></ul>
2	<ul style="list-style-type: none"><li>• PDA</li><li>• kontextfreie Grammatik</li></ul>
3	<ul style="list-style-type: none"><li>• DEA</li><li>• reguläre Grammatik</li><li>• NEA</li><li>• regulärer Ausdruck</li></ul>

# Einordnung

Typ	Beschreibungsmittel
0	<ul style="list-style-type: none"><li>• TM</li><li>• 'Typ-0 Grammatik'</li></ul>
1	<ul style="list-style-type: none"><li>• linear beschränkte TM (LBA)</li><li>• kontextsensitive Grammatik</li></ul>
2	<ul style="list-style-type: none"><li>• PDA</li><li>• kontextfreie Grammatik</li></ul>
3	<ul style="list-style-type: none"><li>• DEA</li><li>• reguläre Grammatik</li><li>• NEA</li><li>• regulärer Ausdruck</li></ul>

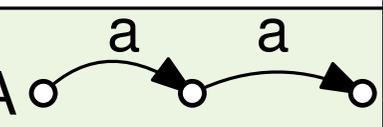
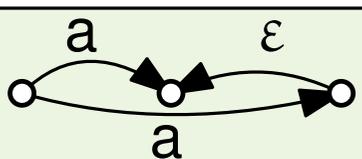
DPDA?

# Einordnung

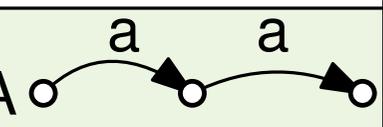
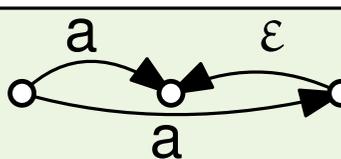
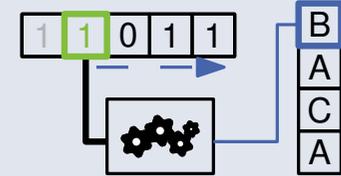
Typ	Beschreibungsmittel
0	<ul style="list-style-type: none"><li>• TM</li><li>• 'Typ-0 Grammatik'</li></ul>
1	<ul style="list-style-type: none"><li>• linear beschränkte TM (LBA)</li><li>• kontextsensitive Grammatik</li></ul>
2	<ul style="list-style-type: none"><li>• PDA</li><li>• kontextfreie Grammatik</li></ul>
deterministisch Kontextfreie Sprachen	
3	<ul style="list-style-type: none"><li>• DEA</li><li>• reguläre Grammatik</li><li>• NEA</li><li>• regulärer Ausdruck</li></ul>

DPDA?

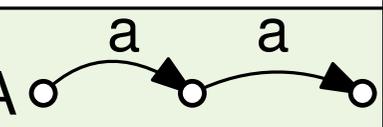
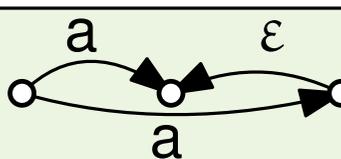
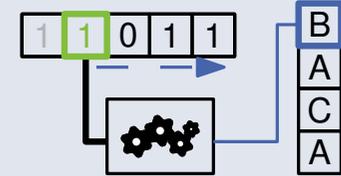
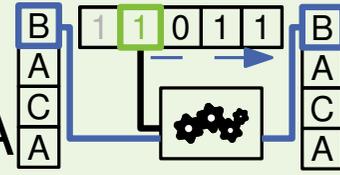
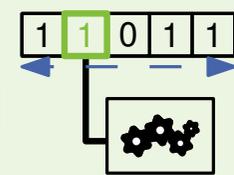
# Maschinenmodelle

		Äquivalent?	Poly. Trafo.?
DEA 	NEA 		

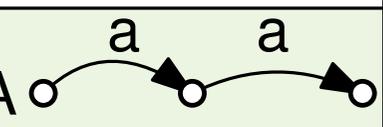
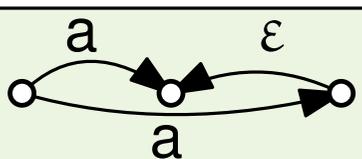
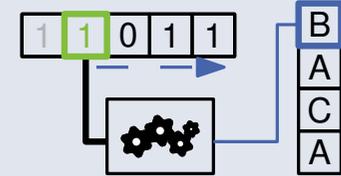
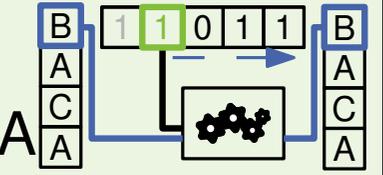
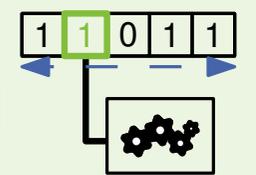
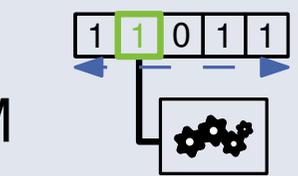
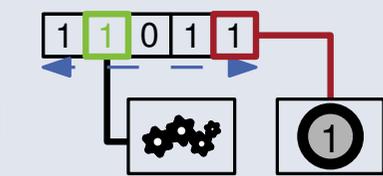
# Maschinenmodelle

		Äquivalent?	Poly. Trafo.?
DEA 	NEA 		
DPDA	PDA 		

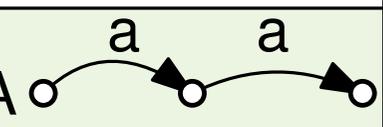
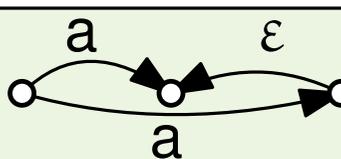
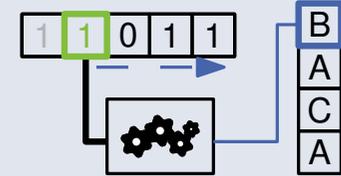
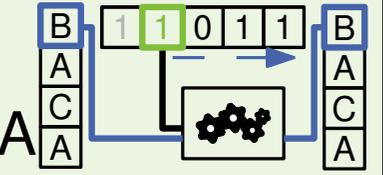
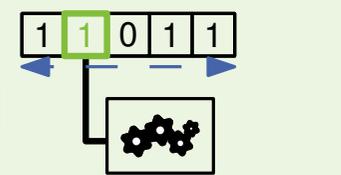
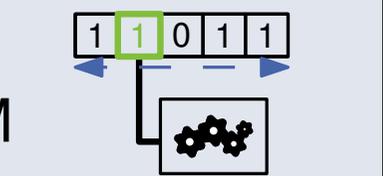
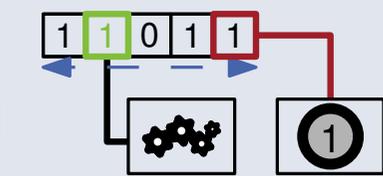
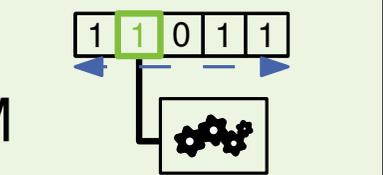
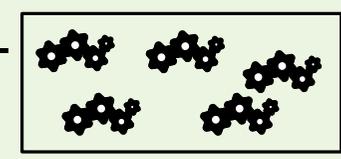
# Maschinenmodelle

		Äquivalent?	Poly. Trafo.?
DEA 	NEA 		
DPDA	PDA 		
2-DPDA 	DTM 		

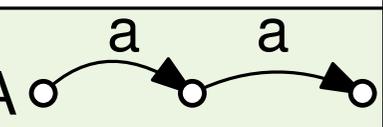
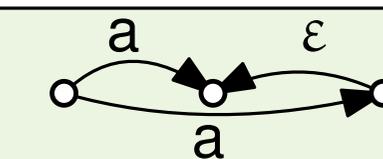
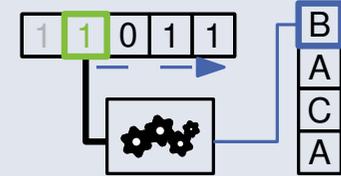
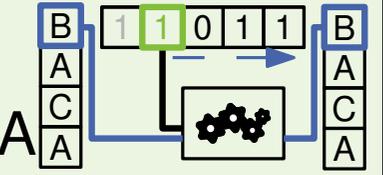
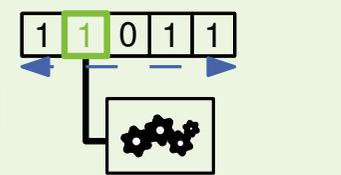
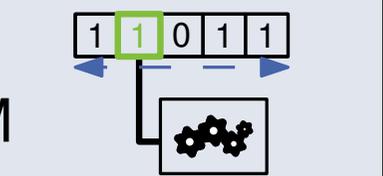
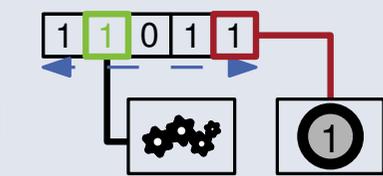
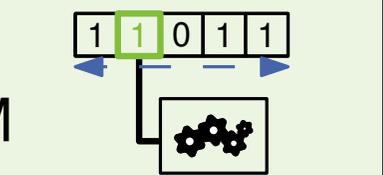
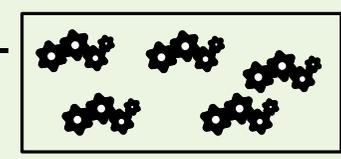
# Maschinenmodelle

		Äquivalent?	Poly. Trafo.?
DEA 	NEA 	✓	✗
DPDA	PDA 	✗	☞
2-DPDA 	DTM 	✓	✓
DTM 	NTM 	✓	$\mathcal{P} = \mathcal{NP}?$

# Maschinenmodelle

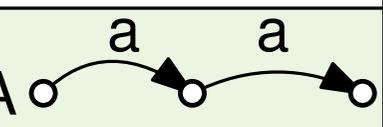
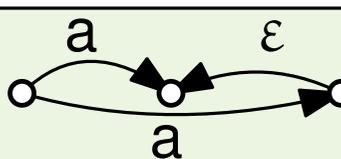
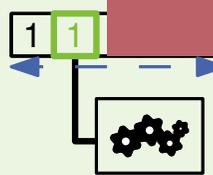
		Äquivalent?	Poly. Trafo.?
DEA 	NEA 	✓	✗
DPDA	PDA 	✗	☞
2-DPDA 	DTM 	✓	✓
DTM 	NTM 	✓	$\mathcal{P} = \mathcal{NP}?$
DTM 	DIE Maschine! 	Intuitiv: ✓	

# Maschinenmodelle

		Äquivalent?	Poly. Trafo.?
DEA 	NEA 	✓	✗
DPDA	PDA 	✗	↔
2-DPDA 	DTM 	✓	✓
DTM 	NTM 	✓	$\mathcal{P} = \mathcal{NP}?$
DTM 	DIE Maschine! 	Intuitiv: ✓	

Mehrband TM, Mehrspur TM, RAM,...

# Maschinenmodelle

		Äquivalent?	Poly. Trafo.?
DEA 	NEA 	✓	✗
DPDA 		✗	
2-DPDA 			✓
DTM 			<i>NP?</i>
DTM 	DTM Maschine!	✓	
Mehrband TM, Mehrspur TM, RAM,...			

Wie werden die Eigenschaften gezeigt?

# Entscheidbarkeit

Typ	$w \in L$	$L(\mathcal{M}) = \emptyset$	$L(\mathcal{M}_1) = L(\mathcal{M}_2)$	$L(\mathcal{M}_1) \cap L(\mathcal{M}_2) = \emptyset$
0	✗	✗	✗	✗
1	✓ $\mathcal{NP}$ -schwer	✗	✗	✗
2	✓ CNF: $O(n^3)$	✓	✗	✗
3	✓ $O(n)$	✓	✓	✓

# Informationstheorie

- Information & Entropie
- verschiedene Kodierungsschemas
- Paritätscodes
- Fehlerkorrektur

# Typ-1 Grammatik



# Typ-1 Grammatik

Grammatik der Form  $\mathbb{X}$ :  $\alpha A \beta \rightarrow \alpha \gamma \beta, \alpha, \beta, \gamma \in (V \cup \Sigma)^*, |\gamma| > 0$

Grammatik der Form  $\mathbb{Y}$ :  $\alpha \rightarrow \beta, |\alpha| \leq |\beta|, \alpha \in V^+, \beta \in ((V \cup \Sigma) \setminus S)^+$

# Typ-1 Grammatik

Grammatik der Form  $\mathbb{X}$ :  $\alpha A \beta \rightarrow \alpha \gamma \beta, \alpha, \beta, \gamma \in (V \cup \Sigma)^*, |\gamma| > 0$

Grammatik der Form  $\mathbb{Y}$ :  $\alpha \rightarrow \beta, |\alpha| \leq |\beta|, \alpha \in V^+, \beta \in ((V \cup \Sigma) \setminus S)^+$

Werden die gleichen Sprachen generiert?

# Typ-1 Grammatik

Grammatik der Form  $\mathbb{X}$ :  $\alpha A \beta \rightarrow \alpha \gamma \beta, \alpha, \beta, \gamma \in (V \cup \Sigma)^*, |\gamma| > 0$

Grammatik der Form  $\mathbb{Y}$ :  $\alpha \rightarrow \beta, |\alpha| \leq |\beta|, \alpha \in V^+, \beta \in ((V \cup \Sigma) \setminus S)^+$

Werden die gleichen Sprachen generiert?

Klar:  $\mathcal{L}(\mathbb{X}) \subset \mathcal{L}(\mathbb{Y})$

$\mathcal{L}(\mathbb{Y}) \subset \mathcal{L}(\mathbb{X})$ :  $X_1 X_2 \dots X_n \rightarrow Y_1 \dots Y_m, m > 1$

# Typ-1 Grammatik

Grammatik der Form  $\mathbb{X}$ :  $\alpha A \beta \rightarrow \alpha \gamma \beta, \alpha, \beta, \gamma \in (V \cup \Sigma)^*, |\gamma| > 0$

Grammatik der Form  $\mathbb{Y}$ :  $\alpha \rightarrow \beta, |\alpha| \leq |\beta|, \alpha \in V^+, \beta \in ((V \cup \Sigma) \setminus S)^+$

Werden die gleichen Sprachen generiert?

Klar:  $\mathcal{L}(\mathbb{X}) \subset \mathcal{L}(\mathbb{Y})$

$\mathcal{L}(\mathbb{Y}) \subset \mathcal{L}(\mathbb{X})$ :  $X_1 X_2 \dots X_n \rightarrow Y_1 \dots Y_m, m > 1$

- $X_1 X_2 \dots X_n \rightarrow Z_1 X_2 \dots X_n$
- $Z_1 X_2 \dots X_n \rightarrow Z_1 Z_2 X_3 \dots X_n$
- ...
- $Z_1 \dots Z_{n-2} X_{n-1} X_n \rightarrow Z_1 \dots Z_{n-2} Z_{n-1} X_n$

# Typ-1 Grammatik

Grammatik der Form  $\mathbb{X}$ :  $\alpha A \beta \rightarrow \alpha \gamma \beta, \alpha, \beta, \gamma \in (V \cup \Sigma)^*, |\gamma| > 0$

Grammatik der Form  $\mathbb{Y}$ :  $\alpha \rightarrow \beta, |\alpha| \leq |\beta|, \alpha \in V^+, \beta \in ((V \cup \Sigma) \setminus S)^+$

Werden die gleichen Sprachen generiert?

Klar:  $\mathcal{L}(\mathbb{X}) \subset \mathcal{L}(\mathbb{Y})$

$\mathcal{L}(\mathbb{Y}) \subset \mathcal{L}(\mathbb{X})$ :  $X_1 X_2 \dots X_n \rightarrow Y_1 \dots Y_m, m > 1$

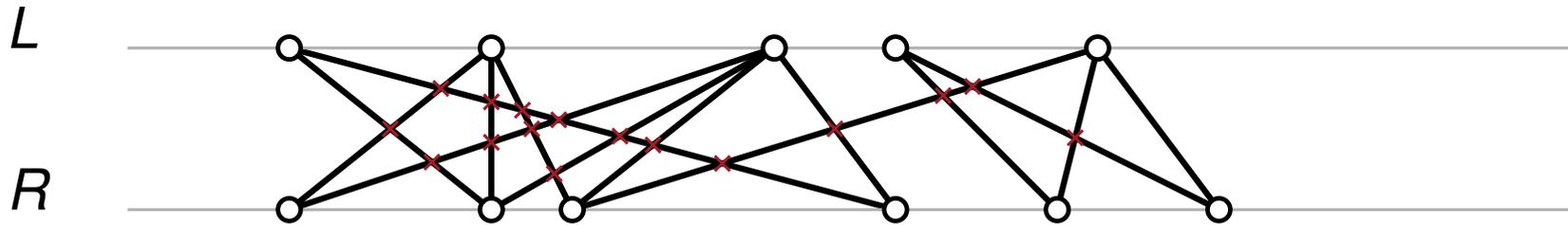
- $Z_1 \dots Z_{n-1} X_n \rightarrow Z_1 \dots Z_{n-1} Z_n Y_{n+1} \dots Y_{n+m}$
- $Z_1 \dots Z_{n-1} Z_n Y_{n+1} \dots Y_{n+m} \rightarrow Y_1 \dots Z_{n-1} Z_n Y_{n+1} \dots Y_{n+m}$
- ...
- $Y_1 \dots Y_{n-1} Z_n Y_{n+1} \dots Y_{n+m} \rightarrow Y_1 \dots Y_{n-1} Y_n Y_{n+1} \dots Y_{n+m}$

# Kreuzungsminimierung



# Einseitige Kreuzungsminimierung (OSCM)

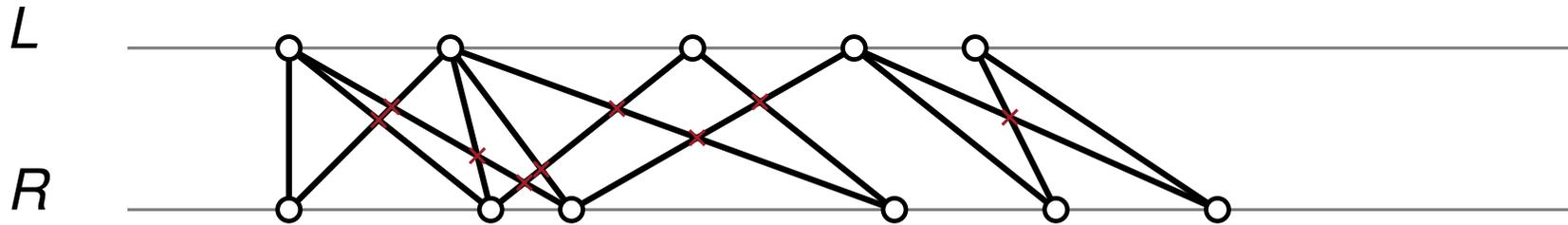
**Geg.:** Bipartiter Graph  $G = (L, R, E)$  und  
Knotenordnung  $r$  von  $R$



**Ges.:** Knotenordnung  $l$  von  $L$ , so dass die Anzahl Kreuzungen von Kanten  
in  $E$  minimal ist

# Einseitige Kreuzungsminimierung (OSCM)

**Geg.:** Bipartiter Graph  $G = (L, R, E)$  und  
Knotenordnung  $r$  von  $R$



**Ges.:** Knotenordnung  $l$  von  $L$ , so dass die Anzahl Kreuzungen von Kanten  
in  $E$  minimal ist

# Einseitige Kreuzungsminimierung (OSCM)

**Geg.:** Bipartiter Graph  $G = (L, R, E)$  und  
Knotenordnung  $r$  von  $R$

**Ges.:** Knotenordnung  $l$  von  $L$ , so dass die Anzahl Kreuzungen von Kanten  
in  $E$  minimal ist

# Einseitige Kreuzungsminimierung (OSCM)

**Geg.:** Bipartiter Graph  $G = (L, R, E)$  und Knotenordnung  $r$  von  $R$

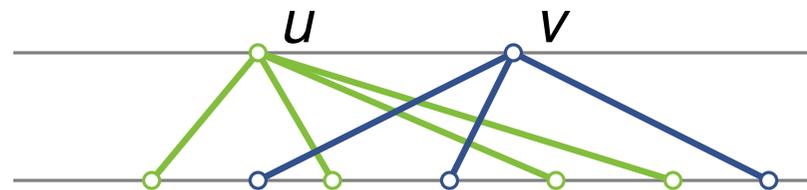
**Ges.:** Knotenordnung  $l$  von  $L$ , so dass die Anzahl Kreuzungen von Kanten in  $E$  minimal ist

## Beobachtung:

- Anzahl Kreuzungen einer 2-Lagen-Zeichnung von  $G$  hängt nur von  $l$  und  $r$  ab, nicht von tatsächlichen Positionen
- für  $u, v \in L$  hängt Anzahl Kreuzungen inzidenter Kanten nur von  $l(u) < l(v)$  oder  $l(v) < l(u)$  ab

**Def:** Kreuzungszahl für  $l(u) < l(v)$

$$c_{uv} := |\{(uw, vz) \mid w \in N(u), z \in N(v), r(z) < r(w)\}|$$



$$c_{uv} = 5$$

$$c_{vu} = 7$$

# Einseitige Kreuzungsminimierung (OSCM)

**Geg.:** Bipartiter Graph  $G = (L, R, E)$  und Knotenordnung  $r$  von  $R$

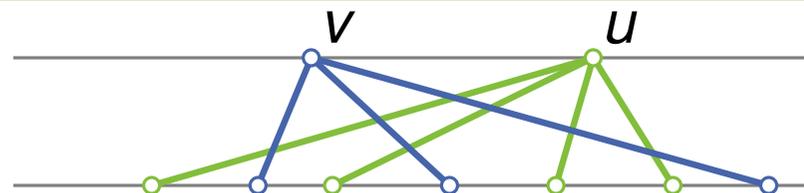
**Ges.:** Knotenordnung  $l$  von  $L$ , so dass die Anzahl Kreuzungen von Kanten in  $E$  minimal ist

## Beobachtung:

- Anzahl Kreuzungen einer 2-Lagen-Zeichnung von  $G$  hängt nur von  $l$  und  $r$  ab, nicht von tatsächlichen Positionen
- für  $u, v \in L$  hängt Anzahl Kreuzungen inzidenter Kanten nur von  $l(u) < l(v)$  oder  $l(v) < l(u)$  ab

**Def:** Kreuzungszahl für  $l(u) < l(v)$

$$c_{uv} := |\{(uw, vz) \mid w \in N(u), z \in N(v), r(z) < r(w)\}|$$



$$c_{uv} = 5$$

$$c_{vu} = 7$$

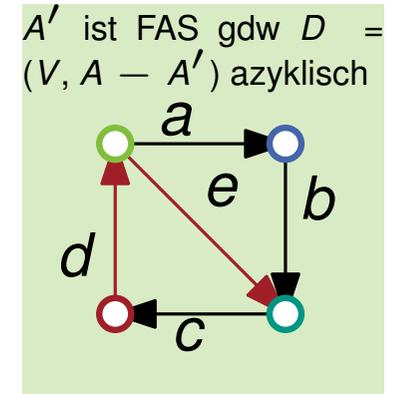
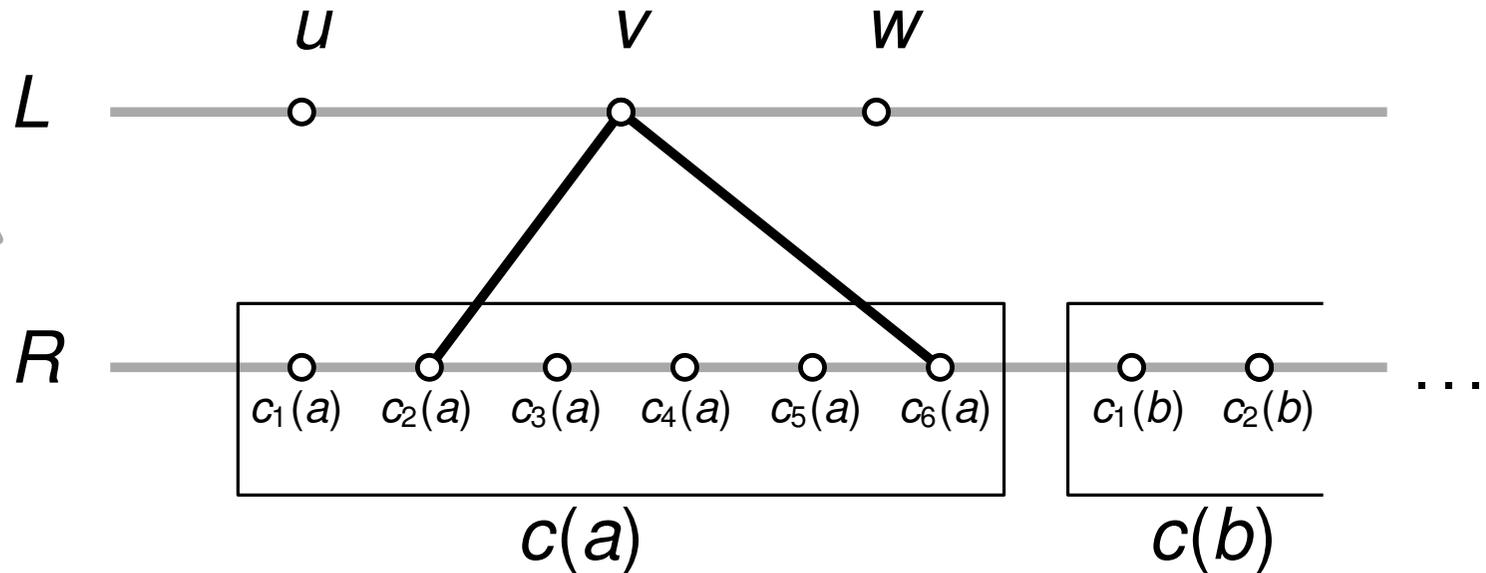
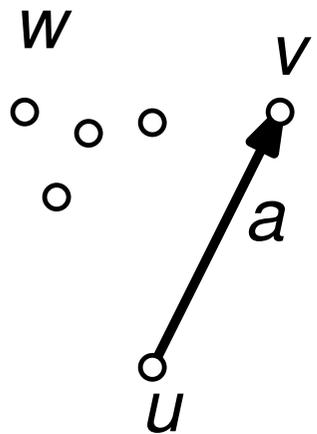
# Transformation

**Satz:** Das einseitige Kreuzungsminimierungsproblem (OSCM) ist  $\mathcal{NP}$ -schwer.

**Reduktion:** FEEDBACK ARC SET

$$D = (V, A)$$

$$B = (W, E)$$



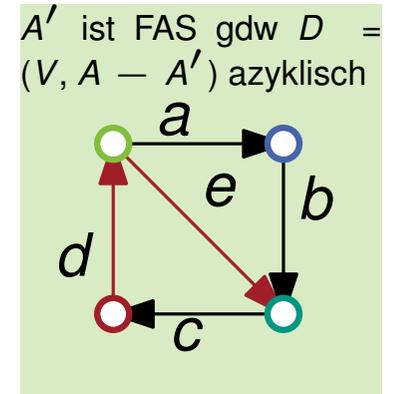
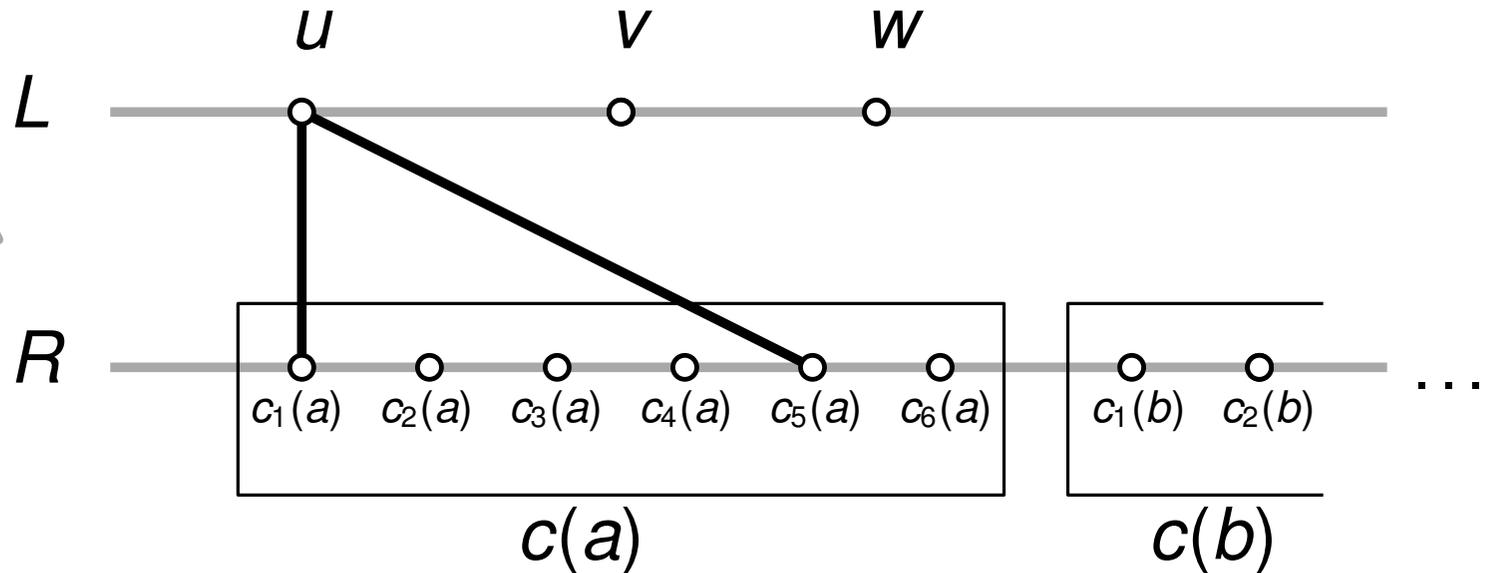
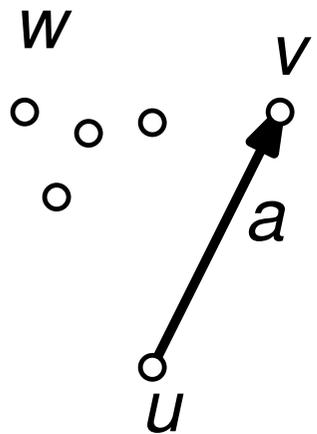
# Transformation

**Satz:** Das einseitige Kreuzungsminimierungsproblem (OSCM) ist  $\mathcal{NP}$ -schwer.

**Reduktion:** FEEDBACK ARC SET

$$D = (V, A)$$

$$B = (W, E)$$



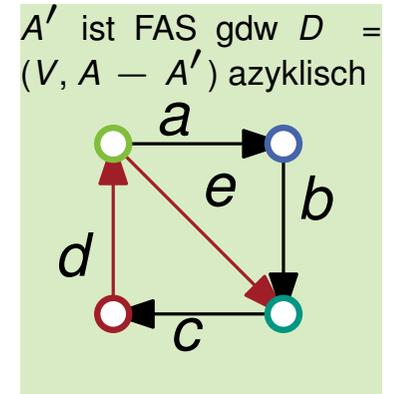
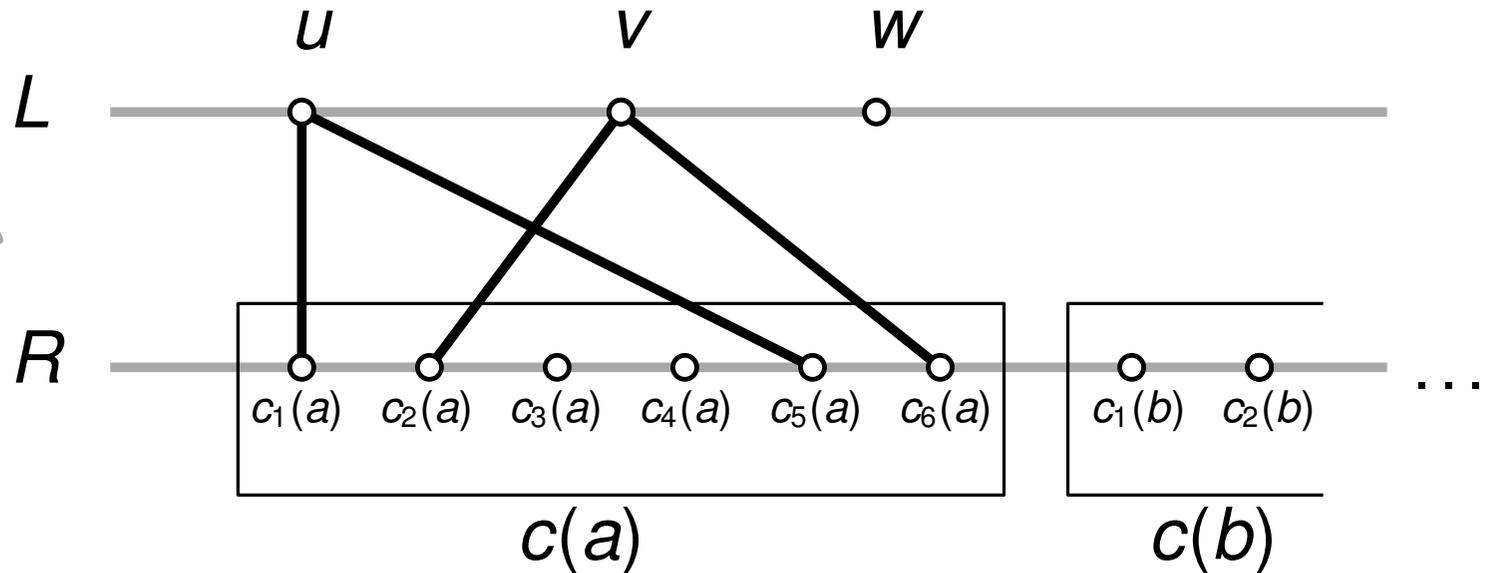
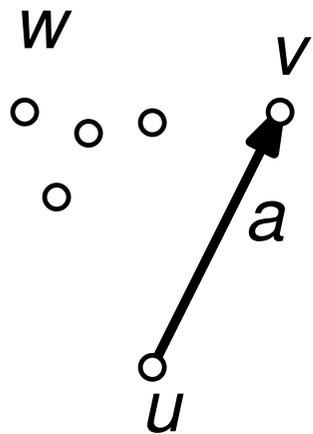
# Transformation

**Satz:** Das einseitige Kreuzungsminimierungsproblem (OSCM) ist  $\mathcal{NP}$ -schwer.

**Reduktion:** FEEDBACK ARC SET

$$D = (V, A)$$

$$B = (W, E)$$



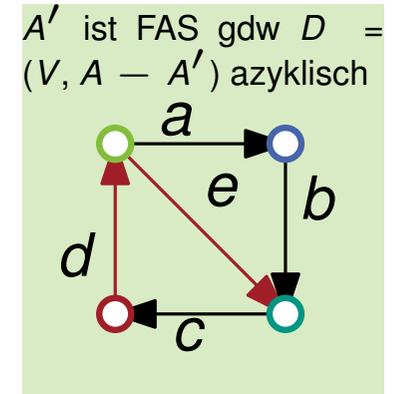
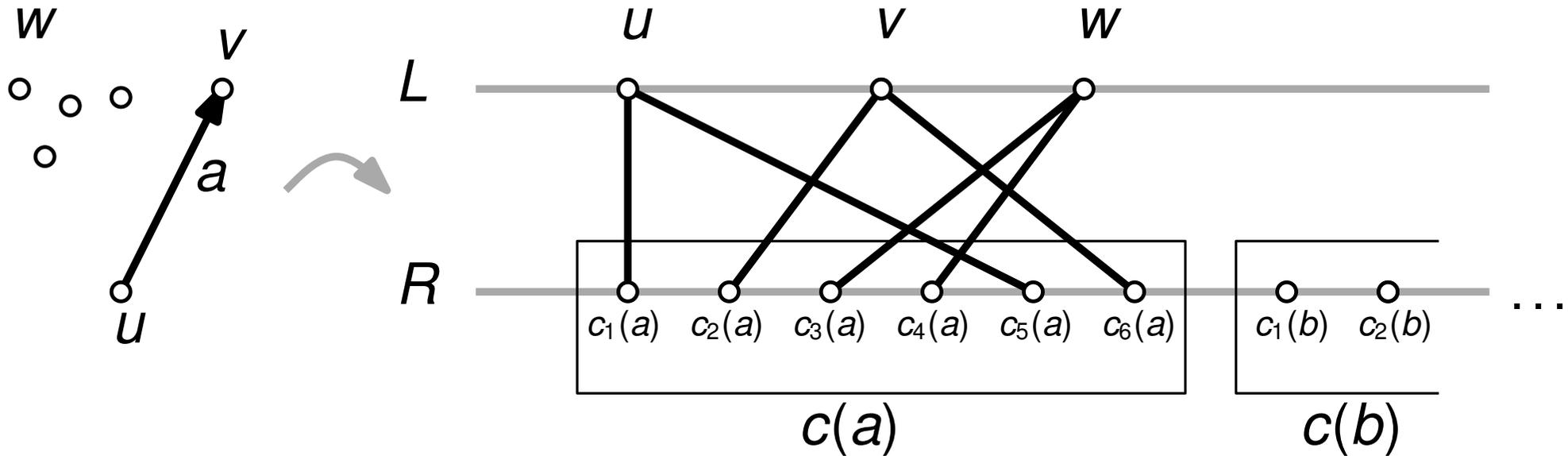
# Transformation

**Satz:** Das einseitige Kreuzungsminimierungsproblem (OSCM) ist  $\mathcal{NP}$ -schwer.

**Reduktion:** FEEDBACK ARC SET

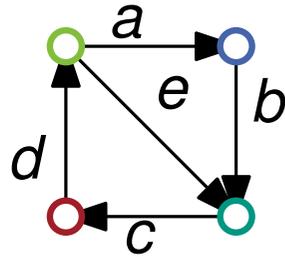
$$D = (V, A)$$

$$B = (W, E)$$

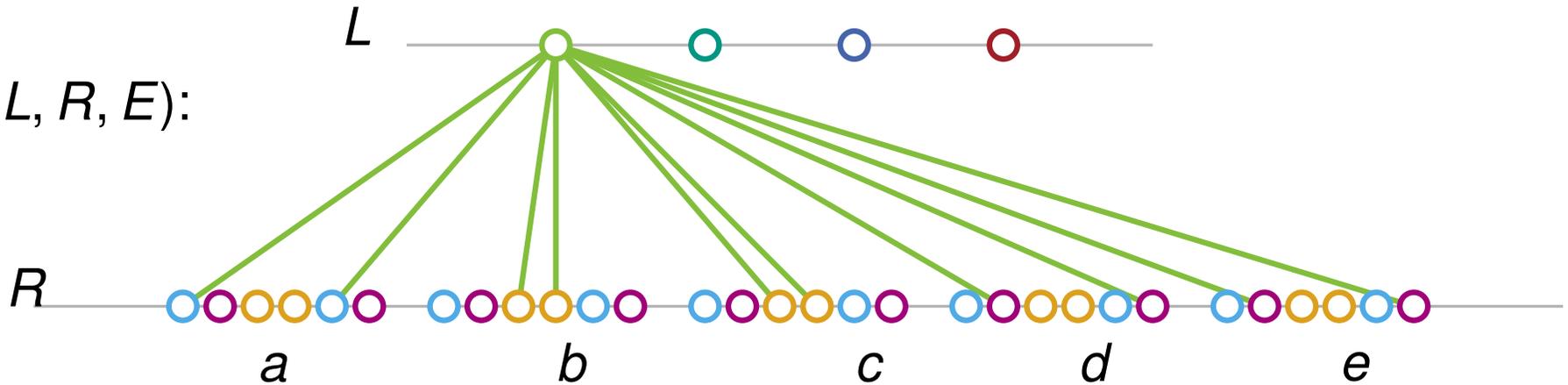


# Beispiel

$D = (V, A)$ :

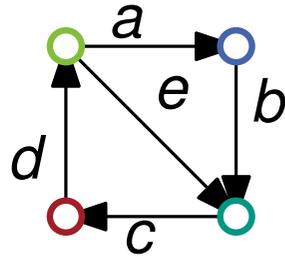


$B = (L, R, E)$ :

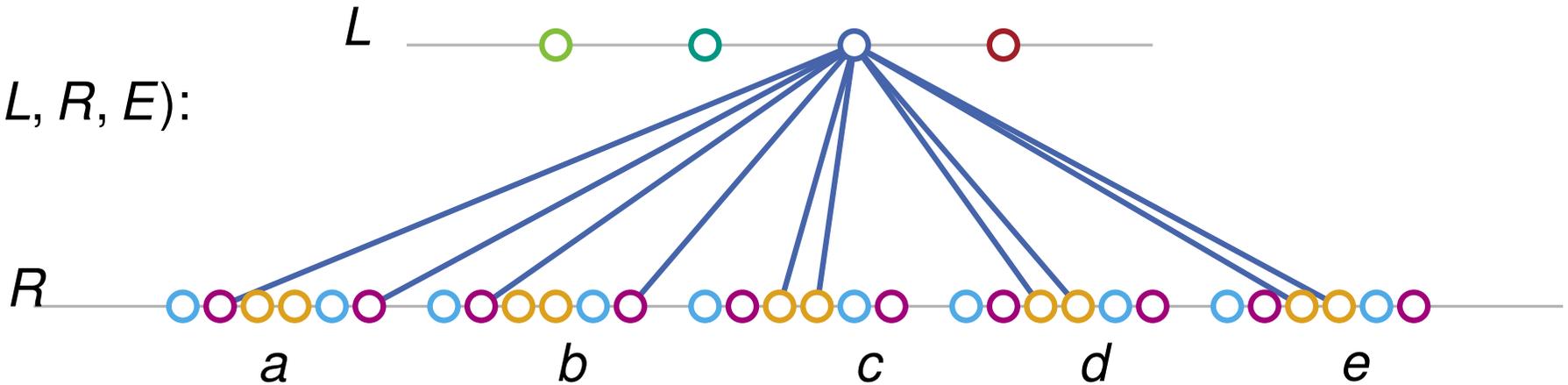


# Beispiel

$D = (V, A)$ :



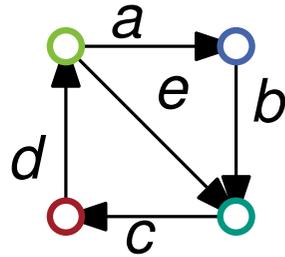
$B = (L, R, E)$ :



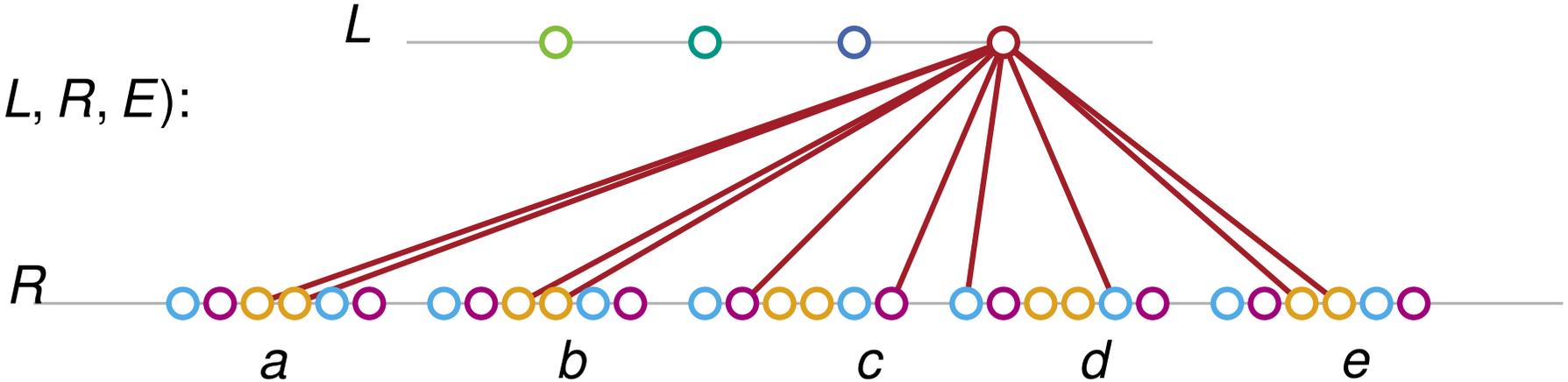


# Beispiel

$D = (V, A)$ :

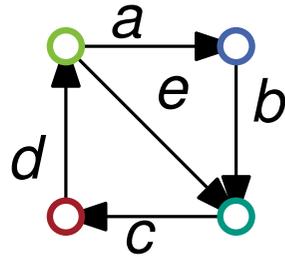


$B = (L, R, E)$ :

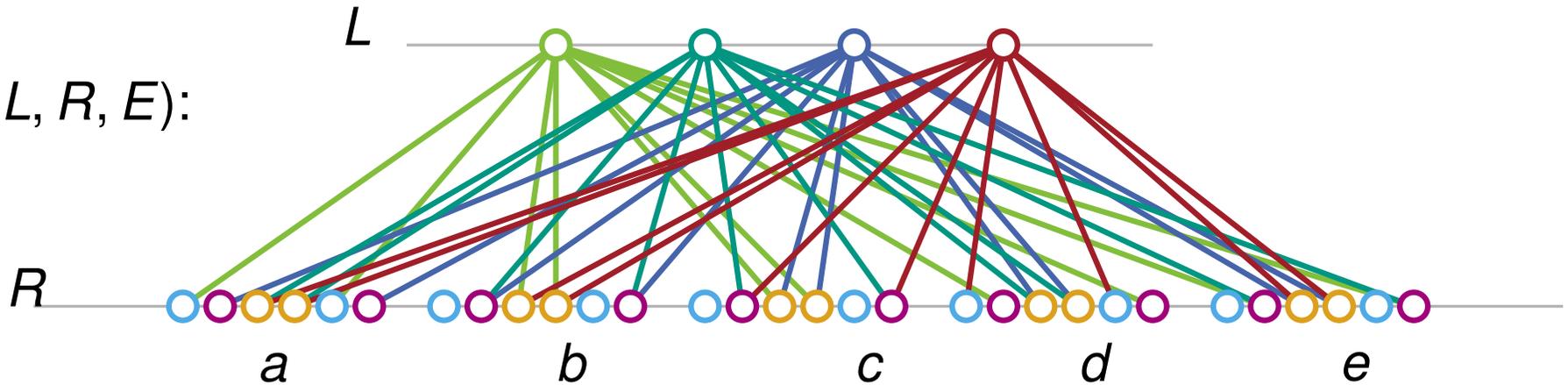


# Beispiel

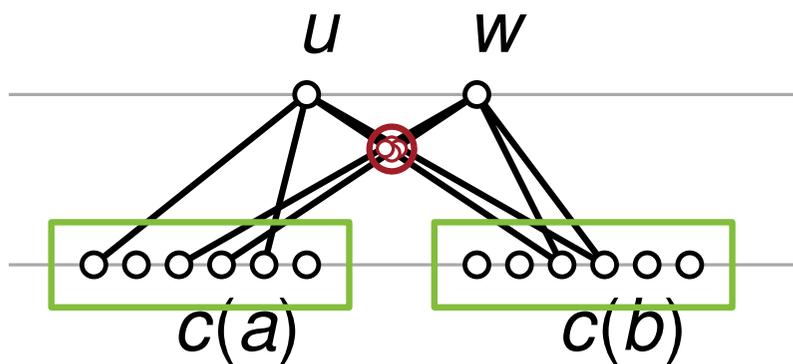
$D = (V, A)$ :



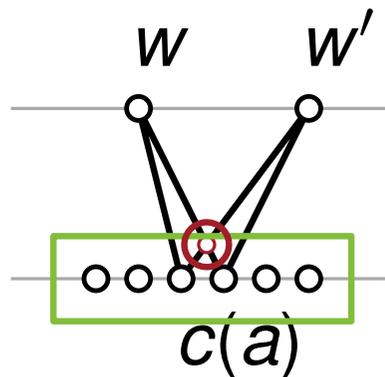
$B = (L, R, E)$ :



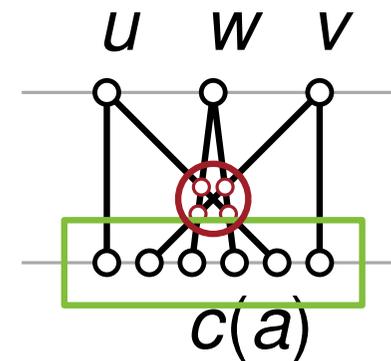
# Kreuzungen Zählen



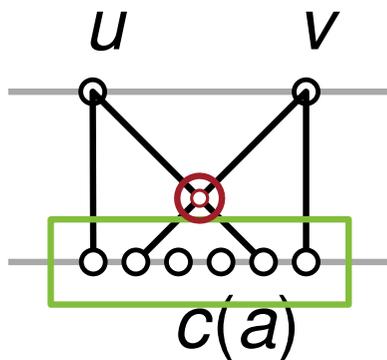
$$4 \cdot \binom{n}{2} \cdot \binom{m}{2}$$



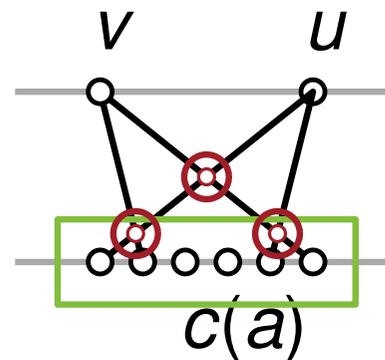
$$+m \cdot \binom{n-2}{2}$$



$$+4 \cdot m \cdot (n - 2)$$



$$+(m - m')$$

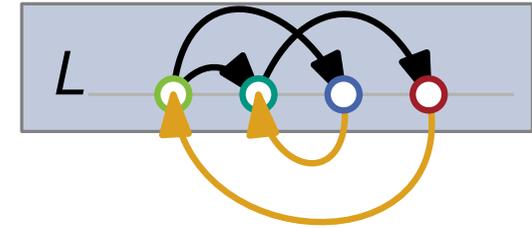
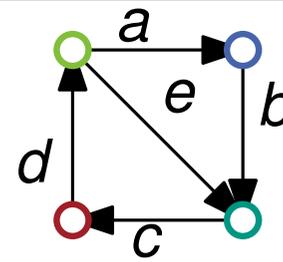


$$+3m'$$

# Korrektheit

$$A_l := \{(u, v) \mid (u, v) \in A, l(u) > l(v)\} = \{b, d\}$$

$$M(x) = 4 \binom{m}{2} \binom{n}{2} + m \binom{n-2}{2} + 4m(n-2) + m + 2x$$

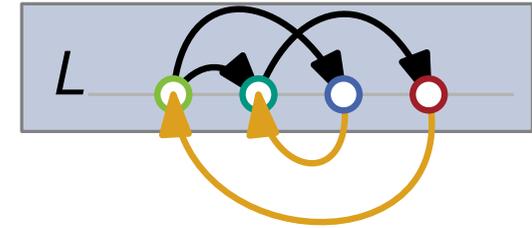
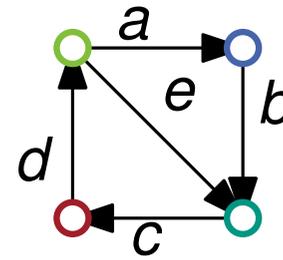


**Lemma:** Sei  $l$  eine Ordnung von  $L$ , dann hat  $B$  genau  $M(|A_l|)$  Kreuzungen

# Korrektheit

$$A_l := \{(u, v) \mid (u, v) \in A, l(u) > l(v)\} = \{b, d\}$$

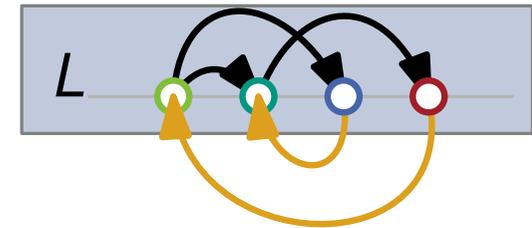
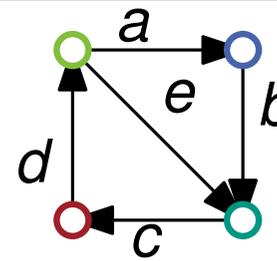
$$M(x) = 4 \binom{m}{2} \binom{n}{2} + m \binom{n-2}{2} + 4m(n-2) + m + 2x$$



**Lemma:** Sei  $l$  eine Ordnung von  $L$ , dann hat  $B$  genau  $M(|A_l|)$  Kreuzungen

$D$  hat FEEDBACK ARC SET der Größe  $K$  gdw.  $B$  hat  $M(K)$  Kreuzungen

# Korrektheit



$$A_l := \{(u, v) \mid (u, v) \in A, l(u) > l(v)\} = \{b, d\}$$

$$M(x) = 4 \binom{m}{2} \binom{n}{2} + m \binom{n-2}{2} + 4m(n-2) + m + 2x$$

**Lemma:** Sei  $l$  eine Ordnung von  $L$ , dann hat  $B$  genau  $M(|A_l|)$  Kreuzungen

$D$  hat FEEDBACK ARC SET der Größe  $K$  gdw.  $B$  hat  $M(K)$  Kreuzungen

Sei  $A'$  ein FAS von  $D$

$D' = (V, A - A')$  azyklisch

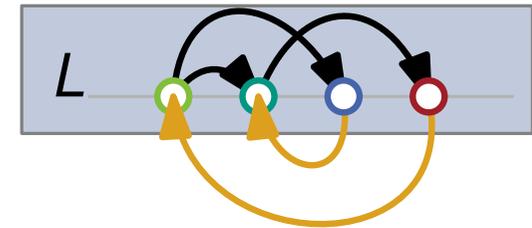
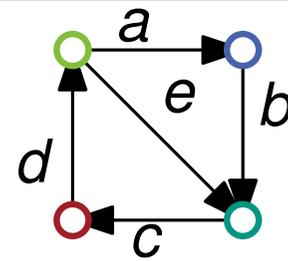
Sei  $l$  eine topologische Sortierung von  $V$  bzgl.  $D'$

$A_l = A'$

$\Rightarrow$  nach Lemma hat  $B$  genau  $M(K)$  Kreuzungen



# Korrektheit



$$A_l := \{(u, v) \mid (u, v) \in A, l(u) > l(v)\} = \{b, d\}$$

$$M(x) = 4 \binom{m}{2} \binom{n}{2} + m \binom{n-2}{2} + 4m(n-2) + m + 2x$$

**Lemma:** Sei  $l$  eine Ordnung von  $L$ , dann hat  $B$  genau  $M(|A_l|)$  Kreuzungen

$D$  hat FEEDBACK ARC SET der Größe  $K$  gdw.  $B$  hat  $M(K)$  Kreuzungen

Sei  $l$  eine Ordnung von  $L$  mit  $M(K)$  Kreuzungen

dann hat  $A_l$  Größe  $K$  (Lemma)

$D' = (V, A - A_l)$  ist azyklisch

$\Rightarrow A_l$  ist FAS der Größe  $K$



# Median Heuristik

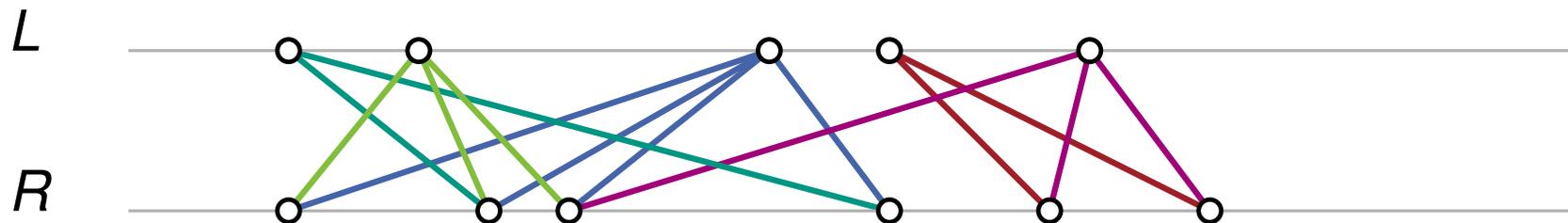
**Idee:** setze Koordinate auf Median der Nachbarn

- für Knoten  $v \in L$  mit Nachbarn  $v_1, \dots, v_k$  setze  $l(v) = \text{med}(v) = r(v_{\lceil k/2 \rceil})$
- falls  $l(u) = l(v)$  mit ungleicher Gradparität, setze ungeraden Knoten nach links
- falls  $l(u) = l(v)$  mit gleicher Gradparität, setze beliebigen Knoten nach links
- Berechnung mit Linearzeit-Medianalgorithmus in  $O(|E|)$

# Median Heuristik

**Idee:** setze Koordinate auf Median der Nachbarn

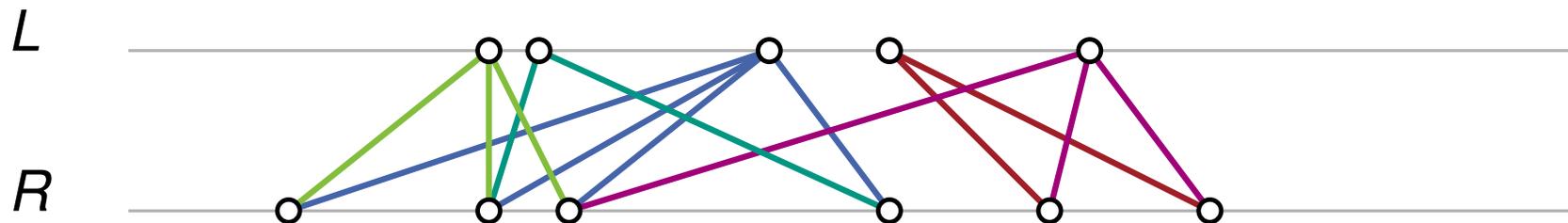
- für Knoten  $v \in L$  mit Nachbarn  $v_1, \dots, v_k$  setze  $l(v) = \text{med}(v) = r(v_{\lceil k/2 \rceil})$
- falls  $l(u) = l(v)$  mit ungleicher Gradparität, setze ungeraden Knoten nach links
- falls  $l(u) = l(v)$  mit gleicher Gradparität, setze beliebigen Knoten nach links
- Berechnung mit Linearzeit-Medianalgorithmus in  $O(|E|)$



# Median Heuristik

**Idee:** setze Koordinate auf Median der Nachbarn

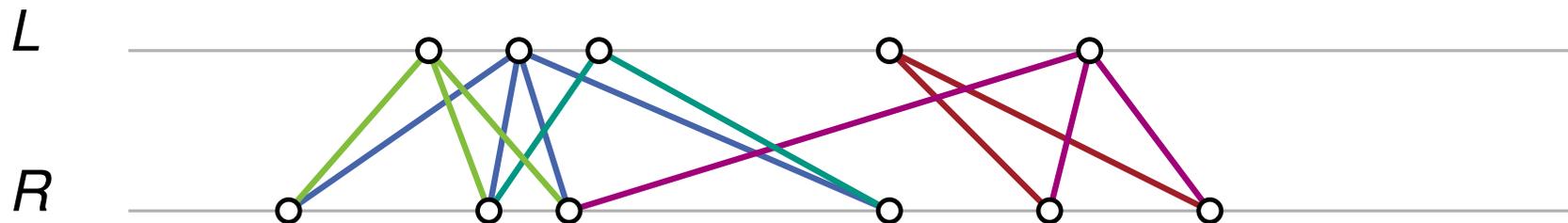
- für Knoten  $v \in L$  mit Nachbarn  $v_1, \dots, v_k$  setze  $l(v) = \text{med}(v) = r(v_{\lceil k/2 \rceil})$
- falls  $l(u) = l(v)$  mit ungleicher Gradparität, setze ungeraden Knoten nach links
- falls  $l(u) = l(v)$  mit gleicher Gradparität, setze beliebigen Knoten nach links
- Berechnung mit Linearzeit-Medianalgorithmus in  $O(|E|)$



# Median Heuristik

**Idee:** setze Koordinate auf Median der Nachbarn

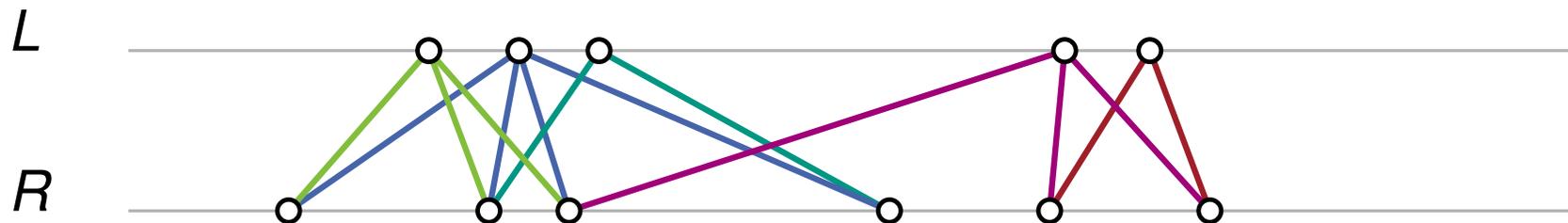
- für Knoten  $v \in L$  mit Nachbarn  $v_1, \dots, v_k$  setze  $l(v) = \text{med}(v) = r(v_{\lceil k/2 \rceil})$
- falls  $l(u) = l(v)$  mit ungleicher Gradparität, setze ungeraden Knoten nach links
- falls  $l(u) = l(v)$  mit gleicher Gradparität, setze beliebigen Knoten nach links
- Berechnung mit Linearzeit-Medianalgorithmus in  $O(|E|)$



# Median Heuristik

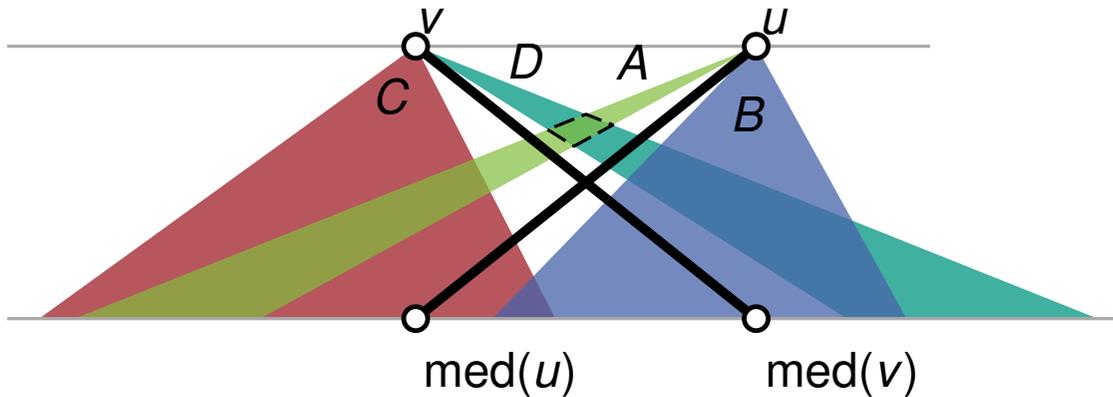
**Idee:** setze Koordinate auf Median der Nachbarn

- für Knoten  $v \in L$  mit Nachbarn  $v_1, \dots, v_k$  setze  $l(v) = \text{med}(v) = r(v_{\lceil k/2 \rceil})$
- falls  $l(u) = l(v)$  mit ungleicher Gradparität, setze ungeraden Knoten nach links
- falls  $l(u) = l(v)$  mit gleicher Gradparität, setze beliebigen Knoten nach links
- Berechnung mit Linearzeit-Medianalgorithmus in  $O(|E|)$

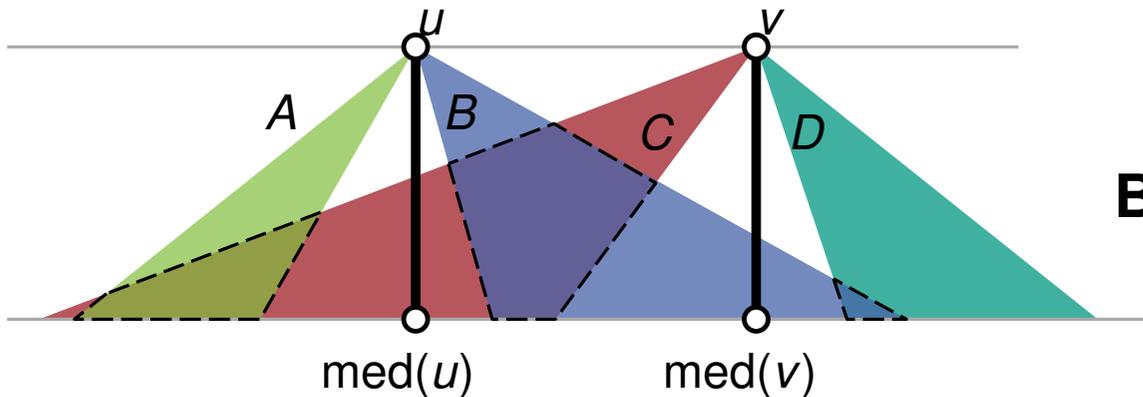


# Approximation

**Satz 2:** Sei  $G = (L, R, E)$  ein bipartiter Graph und  $r$  eine beliebige Ordnung von  $R$ . Dann gilt  $\text{med}(G, r) \leq 3 \text{opt}(G, r)$ .



$$c_{vu} \geq ad + a + d$$



$$c_{uv} \leq ac + bc + bd + c + b$$

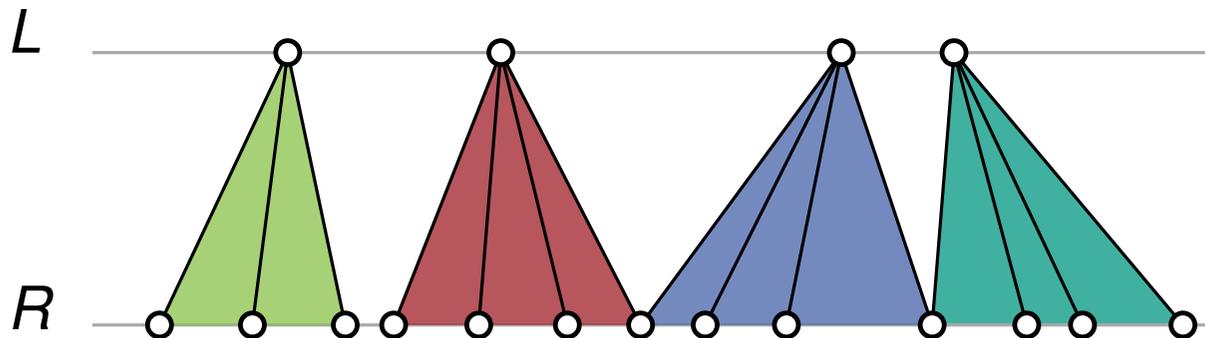
**Beobachtung:**  $b \leq a + 1, c \leq d$

$$\Rightarrow c_{uv} \leq 3ad + 3d + a + 1$$

# Kreuzungsfrei

**Satz 3:** Sei  $G = (L, R, E)$  ein bipartiter Graph und  $r$  eine beliebige Ordnung von  $R$  mit  $\text{opt}(G, r) = 0$ . Dann gilt  $\text{med}(G, r) = 0$ .

**Beobachtung:** Nachbarschaften der Knoten aus  $L$  bilden (fast) disjunkte Intervalle bzgl.  $r$



# Reduktionsschemas



# Reduktionsschemas

## Beschränkung eines Problems

- offensichtliche 1 zu 1 Beziehung
- $3SAT \propto 4SAT$

## Lokales Ersetzen

- Veränderung der lokalen Struktur
- weitestgehend unabhängige Veränderung
- $3SAT \propto MAX2SAT$

## Komponenten Design

- Modulierung von Interaktion zwischen Komponenten
- Modeliere z.B. Literale und Klauseln
- $3SAT \propto 3COLOR$

# Beschränkung

## Problem 3SAT:

**Gegeben:** Menge  $U$  von Variablen, Menge  $C_3$  von Klauseln über  $U$ , wobei jede Klausel genau **drei** Literale enthält

**Frage:** Existiert eine Wahrheitbelegung die alle Klauseln erfüllt?

## Problem 4SAT:

**Gegeben:** Menge  $U$  von Variablen, Menge  $C_4$  von Klauseln über  $U$ , wobei jede Klausel genau **vier** Literale enthält

**Frage:** Existiert eine Wahrheitbelegung die alle Klauseln erfüllt?

# Beschränkung

## Problem 3SAT:

**Gegeben:** Menge  $U$  von Variablen, Menge  $C_3$  von Klauseln über  $U$ , wobei jede Klausel genau **drei** Literale enthält

**Frage:** Existiert eine Wahrheitbelegung die alle Klauseln erfüllt?

## Problem 4SAT:

**Gegeben:** Menge  $U$  von Variablen, Menge  $C_4$  von Klauseln über  $U$ , wobei jede Klausel genau **vier** Literale enthält

**Frage:** Existiert eine Wahrheitbelegung die alle Klauseln erfüllt?

## Transformation 3SAT $\propto$ 4SAT

Sei  $(C_3, U)$  eine 3SAT Instanz

$$(C_4 = \{a \vee b \vee c \vee d \mid a \vee b \vee c \in C_3, U \cup \{d\})$$

# Lokale Ersetzung

## Problem EXACT COVER BY 3SAT (X3C):

**Gegeben:** Endliche Menge  $X$  mit  $|X| = 3q$  und Menge  $C$  von 3 elementigen Teilmengen von  $X$ .

**Frage:** Existiert eine exakte Überdeckung von  $X$  mit Mengen aus  $C$ ?

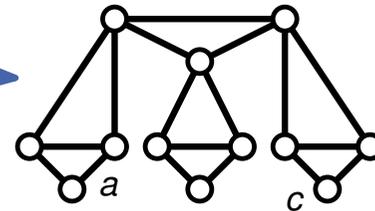
## Problem PARTITION INTO TRIANGLES (PIT)

**Gegeben:** Ungerichteter Graph  $G = (V, E)$  mit  $|V| = 3q$ .

**Frage:** Gibt es eine Partitionierung  $V_1 \cup V_2 \cup \dots \cup V_k$  von  $V$ , so dass jedes  $V_i$  ein Dreieck in  $G$  induziert?

## Transformation X3C $\propto$ PIT

$$C = \{ \{a, b, c\} \}$$



# Lokale Ersetzung

## Problem EXACT COVER BY 3SAT (X3C):

**Gegeben:** Endliche Menge  $X$  mit  $|X| = 3q$  und Menge  $C$  von 3 elementigen Teilmengen von  $X$ .

**Frage:** Existiert eine exakte Überdeckung von  $X$  mit Mengen aus  $C$ ?

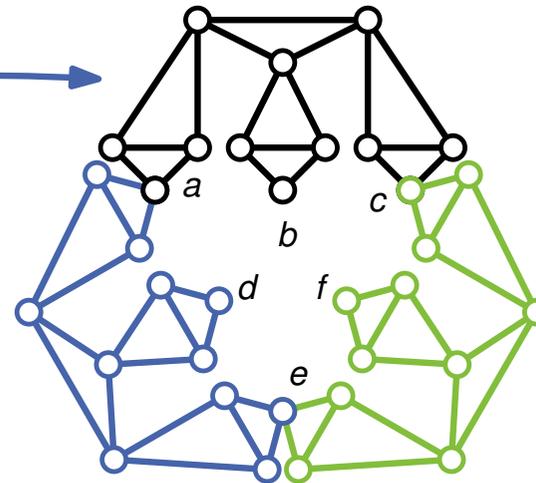
## Problem PARTITION INTO TRIANGLES (PIT)

**Gegeben:** Ungerichteter Graph  $G = (V, E)$  mit  $|V| = 3q$ .

**Frage:** Gibt es eine Partitionierung  $V_1 \cup V_2 \cup \dots \cup V_k$  von  $V$ , so dass jedes  $V_i$  ein Dreieck in  $G$  induziert?

## Transformation X3C $\propto$ PIT

$$C = \{ \{a, b, c\}, \{a, d, e\}, \{c, f, e\} \}$$

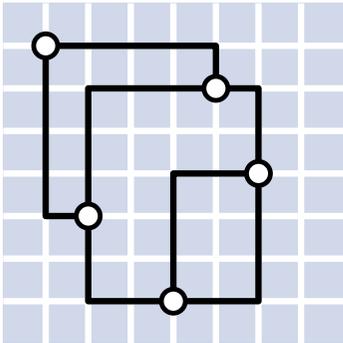


# Komponenten Design

Problem ORTHOGONALE ZEICHNUNG

**Gegeben:** Ungerichteter planarer Graph  $G = (V, E)$ ,  $K \in \mathbb{N}$ .

**Frage:** Existiert eine orthogonale Zeichnung von  $G$  auf einem Gitter der Größe  $K$ ?



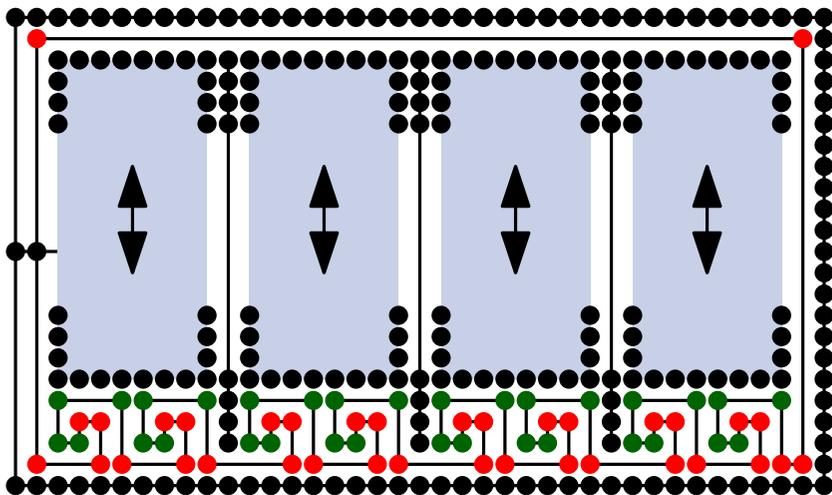
# Komponenten Design

Problem ORTHOGONALE ZEICHNUNG

**Gegeben:** Ungerichteter planarer Graph  $G = (V, E)$ ,  $K \in \mathbb{N}$ .

**Frage:** Existiert eine orthogonale Zeichnung von  $G$  auf einem Gitter der Größe  $K$ ?

Transformation  $\text{SAT} \propto \text{ORTHOGONALE ZEICHNUNG}$



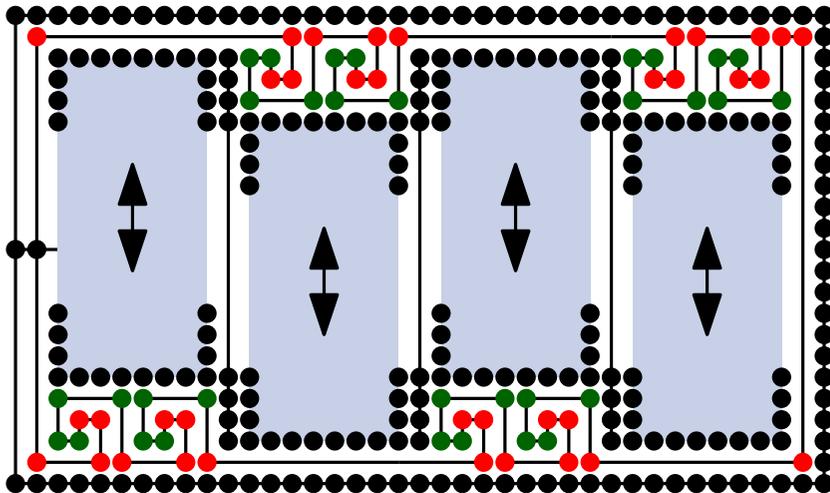
# Komponenten Design

Problem ORTHOGONALE ZEICHNUNG

**Gegeben:** Ungerichteter planarer Graph  $G = (V, E)$ ,  $K \in \mathbb{N}$ .

**Frage:** Existiert eine orthogonale Zeichnung von  $G$  auf einem Gitter der Größe  $K$ ?

Transformation  $\text{SAT} \propto \text{ORTHOGONALE ZEICHNUNG}$



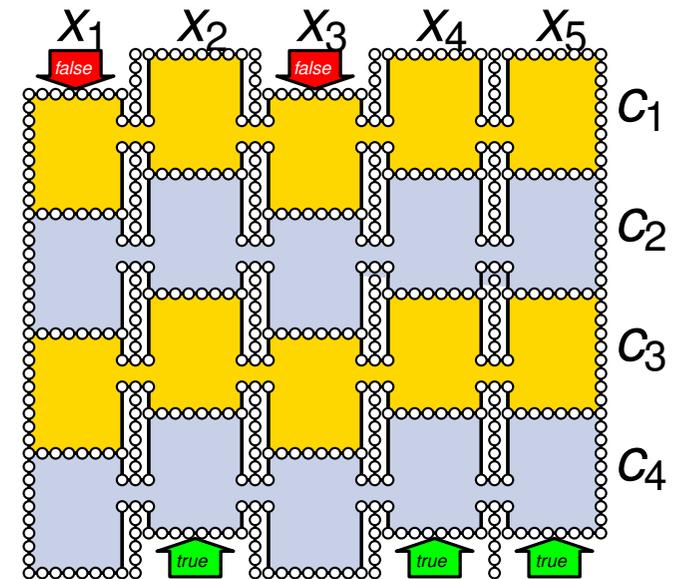
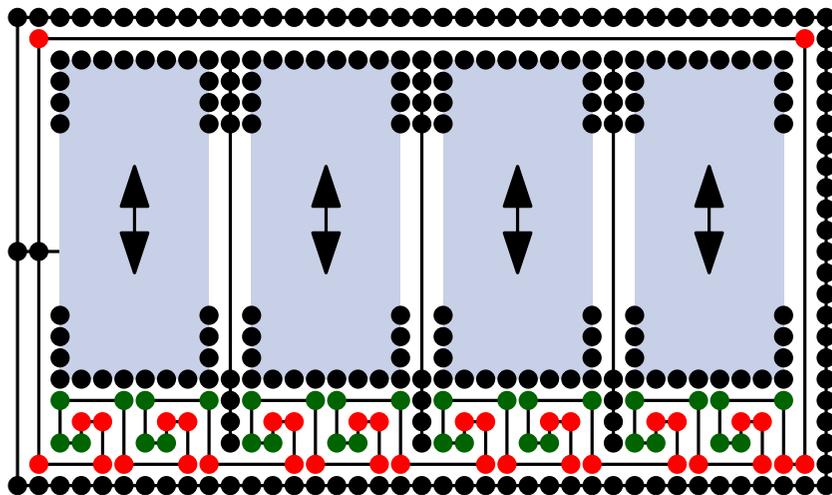
# Komponenten Design

Problem ORTHOGONALE ZEICHNUNG

**Gegeben:** Ungerichteter planarer Graph  $G = (V, E)$ ,  $K \in \mathbb{N}$ .

**Frage:** Existiert eine orthogonale Zeichnung von  $G$  auf einem Gitter der Größe  $K$ ?

Transformation  $\text{SAT} \propto \text{ORTHOGONALE ZEICHNUNG}$



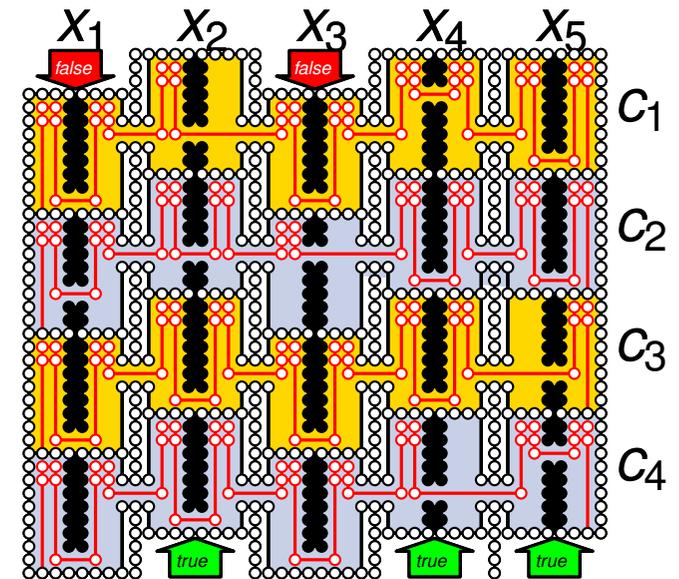
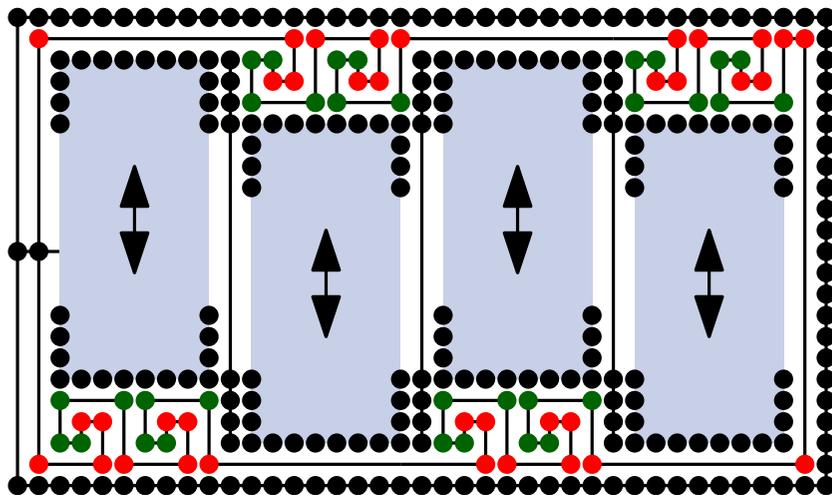
# Komponenten Design

Problem ORTHOGONALE ZEICHNUNG

**Gegeben:** Ungerichteter planarer Graph  $G = (V, E)$ ,  $K \in \mathbb{N}$ .

**Frage:** Existiert eine orthogonale Zeichnung von  $G$  auf einem Gitter der Größe  $K$ ?

Transformation  $\text{SAT} \propto \text{ORTHOGONALE ZEICHNUNG}$



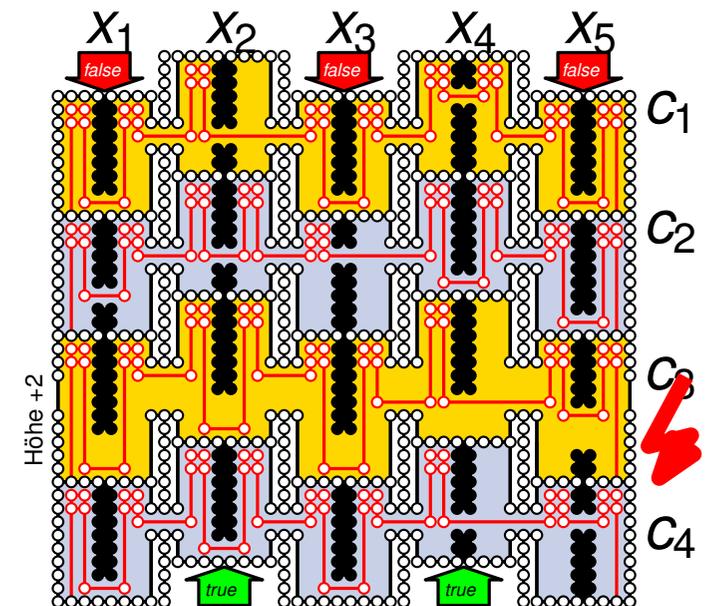
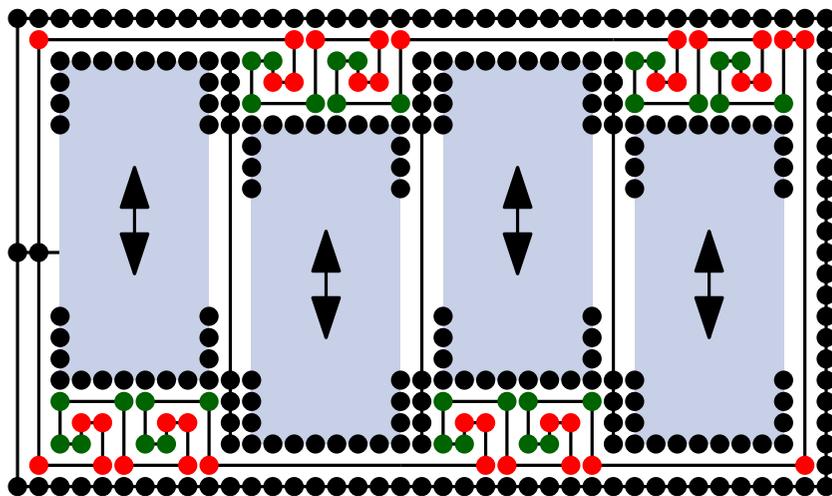
# Komponenten Design

Problem ORTHOGONALE ZEICHNUNG

**Gegeben:** Ungerichteter planarer Graph  $G = (V, E)$ ,  $K \in \mathbb{N}$ .

**Frage:** Existiert eine orthogonale Zeichnung von  $G$  auf einem Gitter der Größe  $K$ ?

Transformation  $\text{SAT} \propto \text{ORTHOGONALE ZEICHNUNG}$



# Klausurhinweise



27.03., Montag

Letzte Möglichkeit zur Anmeldung.

30.03., Freitag

31.03., Dienstag

Bekanntgabe der Hörsaalverteilung:

- **Rechtzeitig** überprüfen, ob man auf der Liste steht.
- Hörsaal herausfinden, in dem man schreibt.

30.04., Montag  
Klausur

27.03., Montag

Letzte Möglichkeit zur Anmeldung.

30.03., Freitag

31.03., Dienstag

Bekanntgabe der Hörsaalverteilung:

- **Rechtzeitig** überprüfen, ob man auf der Liste steht.
- Hörsaal herausfinden, in dem man schreibt.

30.04., Montag  
Klausur

### Ablauf der Klausur:

- Klausur beginnt **pünktlich** um 11:00 Uhr
- Bitte **frühzeitig** im **zugewiesenen** Hörsaal einfinden.
- Bearbeitungszeit: 2 Stunden.

27.03., Montag

Letzte Möglichkeit zur Anmeldung.

30.03., Freitag

31.03., Dienstag

Bekanntgabe der Hörsaalverteilung:

- **Rechtzeitig** überprüfen, ob man auf der Liste steht.
- Hörsaal herausfinden, in dem man schreibt.

30.04., Montag  
Klausur

**Nicht vergessen:**

- Studentenausweis
- Stifte
- Personalausweis!

### Ablauf der Klausur:

- Klausur beginnt **pünktlich** um 11:00 Uhr
- Bitte **frühzeitig** im **zugewiesenen** Hörsaal einfinden.
- Bearbeitungszeit: 2 Stunden.

27.03., Montag

Letzte Möglichkeit zur Anmeldung.

30.03., Freitag

31.03., Dienstag

Bekanntgabe der Hörsaalverteilung:

- **Rechtzeitig** überprüfen, ob man auf der Liste steht.
- Hörsaal herausfinden, in dem man schreibt.

30.04., Montag  
Klausur

**Nicht vergessen:**

- Studentenausweis
- Stifte
- Personalausweis!

### Ablauf der Klausur:

- Klausur beginnt **pünktlich** um 11:00 Uhr
- Bitte **frühzeitig** im **zugewiesenen** Hörsaal erscheinen
- Bearbeitungszeit: 2 Stunden.

**Keine Hilfsmittel!**

27.03., Montag

Letzte Möglichkeit zur Anmeldung.

30.03., Freitag

Bekanntgabe der Hörsaalverteilung:

- **Rechtzeitig** überprüfen, ob man auf der Liste steht

### Abmeldung:

- Bis 27.03.17: über Studierendenportal
- Bis 31.03.17: formloses Schreiben bei Übungsleiter oder Sekretariat abgeben (Studentenausweis mit bringen!).
- Am 03.04.17: Nur noch **direkt vor** der Klausur im entsprechend Hörsaal (Studentenausweis mit bringen!).
- Klausur beginnt **pünktlich** um 11:00 Uhr
- Bitte **frühzeitig** im **zugewiesenen** Hörsaal einfinden.
- Bearbeitungszeit: 2 Stunden.

# Klausur

**Inhalt der Klausur:** Stoff der Vorlesung und Übung (ohne Einschränkung)

**Vorsicht:** Übungsleiter können nicht für Korrektheit der Tutorien garantieren!

## Allgemeine Tipps zur Klausur:

- Hinweise beachten!
- Mit Aufgaben anfangen, die man beherrscht.
- Falls Ankreuzaufgaben gestellt werden:
  - ausreichend Zeit nehmen
  - auf Formulierungen achten.
  - Hilfreich: Gegenbeispiele, Proberechnungen und Argumente.

# Leicht vermeidbare Fehler

## Aufgabenstellung:

Geben Sie einen (Keller-)Automaten,  
eine Turingmaschine, eine Grammatik an . . .

Definition der einzelnen Bestandteile nicht vergessen!!!

- Angabe als explizite Liste oder als Tupel.
- Wichtig ist, dass klar wird, wie zum Beispiel bei Grammatiken die Terminal- und Variablenmenge aussieht: Die Angabe der Produktionen alleine genügt nicht!!!
- Wenn **vollständiger** Automat verlangt: Fehlerzustand nicht vergessen.

# Leicht vermeidbare Fehler

## Aufgabenstellung:

Geben Sie einen (Keller-)Automaten,  
eine Turingmaschine, eine Grammatik an . . .

Definition der einzelnen Bestandteile nicht vergessen!!!

- Angabe als explizite Liste oder als Tupel.
- Wichtig ist, dass klar wird, wie zum Beispiel bei Grammatiken die Terminal- und Variablenmenge aussieht: Die Angabe der Produktionen alleine genügt nicht!!!
- Wenn **vollständiger** Automat verlangt: Fehlerzustand nicht vergessen.

## Aufgabenstellung:

- Zeigen Sie, dass . . .
- Zeigen oder widerlegen Sie, dass . . .

Schreiben Sie, was Sie zeigen möchten: "Die Aussage gilt / gilt nicht".

Antwort ausreichend begründen!

# Leicht vermeidbare Fehler

## Aufgabenstellung:

Geben Sie einen (Keller-)Automaten,  
eine Turingmaschine, eine Grammatik an . . .

Definition der einzelnen Bestandteile nicht vergessen!!!

- Angabe als explizite Liste oder als Tupel.
- Wichtig ist, dass klar wird, wie zum Beispiel bei Grammatiken die Terminal- und Variablenmenge aussieht: Die Angabe der Produktionen alleine genügt nicht!!!
- Wenn **vollständiger** Automat verlangt: Fehlerzustand nicht vergessen.

## Aufgabenstellung:

- Zeigen Sie, dass . . .
- Zeigen oder widerlegen Sie, dass . . .

Schreiben Sie, was Sie zeigen möchten: "Die Aussage gilt / gilt nicht".

Antwort ausreichend begründen!

## Aufgabenstellung:

- Berechnen Sie mithilfe des Verfahrens aus der Vorlesung . . .

Schritte der Berechnung dokumentieren!

# Leicht vermeidbare Fehler

## Aufgabenstellung:

Beweisen Sie, dass . . .

Wenn nicht anders verlangt, alles notwendige beweisen!

**Beispiel:** Beweise, dass Problem NP-vollständig ist.

Nicht nur zeigen, dass Problem NP-schwer ist, sondern auch, dass es in NP liegt.

# Viel Erfolg bei der Klausur!



Danke für die hübschen Dr.  
Meta - Zeichnungen!

